

CLAIMS

What is claimed is:

1. A method of verifying the configuration of a switching fabric that interconnects a plurality of end nodes into a cluster, the switching fabric including at least one switch and a plurality of links, each interconnected end node having a fabric management process, each switch having a plurality of ports, the method comprising:

obtaining, from the switch, stored information and saving the stored information so as to be accessible to the fabric management process, the stored information being gathered by said switch;

determining from the stored information whether or not a link is connected to any one of said switch ports;

for each switch port having a link connected thereto, determining whether the stored information gathered by said switch and pertaining to each said switch port is valid;

for each switch port for which the gathered information is determined to be valid, performing a plurality of tests on the gathered information pertaining to each said switch port;

if the gathered information pertaining to each said switch port passes each test in at least a subset of the plurality of tests, enabling each said switch port for data traffic; and

otherwise, disabling each said switch port for data traffic.

2. A method of verifying as recited in claim 1, wherein the stored information is gathered periodically by the switch.

3. A method of verifying as recited in claim 1,

wherein any neighboring port connected to any switch port belongs to an expected fabric; and

wherein the plurality of tests on the gathered information includes a test to determine whether the neighboring port belongs to the expected fabric.

4. A method of verifying as recited in claim 1,

wherein any neighboring port connected to any switch port has an expected port number;
and

wherein the plurality of tests on the gathered information includes a test to determine whether the neighboring port has the expected port number.

5

5. A method of verifying as recited in claim 1,

wherein any switch having a port connected to said switch port has a global unique identification (GUID) number; and

10 wherein the plurality of tests on the gathered information includes a test to determine whether the global unique identification of the neighboring port has a valid format.

6. A method of verifying as recited in claim 1,

wherein any switch, having a port connected to said switch port, has a configuration version id, a configuration tag and a manufacturing part number; and

wherein the plurality of tests on the gathered information includes a test to determine whether the configuration version id, configuration tag and manufacturing part number of the switch connected to said switch port have valid formats.

7. A method of verifying as recited in claim 1, wherein there is a single bundle in the switching fabric, said bundle including two or more links that interconnect neighboring switches.

8. A method of verifying as recited in claim 7,

wherein any neighboring port connected to said switch port is part of the bundle, each port in the bundle having the same GUID; and

25 wherein the plurality of tests on the gathered information includes a test to determine the GUID of the neighboring port connected to said switch port to determine whether the neighboring port is properly in the bundle.

9. A method of verifying as recited in claim 1, wherein there are at least two bundles in the switching fabric, each said bundle including two or more links that interconnect neighboring switches.

10. A method of verifying as recited in claim 7,

wherein any neighboring port connected to said switch port is part of the first bundle, each port in the bundle having the same GUID; and

wherein the plurality of tests on the gathered information includes a test to determine the GUID of the neighboring port connected to said switch port to determine whether the neighboring port is properly in the first bundle and not in the second bundle.

11. A method of verifying as recited in claim 1,

wherein any switch, having a port connected to said switch port, has a configuration tag to uniquely specify an expected cluster topology ID and an expected position ID of the switch therein; and

wherein the plurality of tests on the gathered information includes a test to determine from the configuration tag for the switch connected to said switch port whether the switch is configured for the expected cluster topology and is in the expected position in the cluster topology.

12. A method of verifying as recited in claim 1,

wherein any switch, having a port connected to said switch port, has a firmware release revision, firmware major revision, firmware minor revision, configuration major revision, and configuration minor revision; and

wherein the plurality of tests on the gathered information includes a test to determine whether the firmware release revision, firmware major revision, firmware minor revision, configuration major revision, and configuration minor revision of the switch connected to said switch port are equal to or higher than a minimum specified level for compatibility.

13. An end node interconnected by a switching fabric to a plurality of other end nodes of a cluster, the switching fabric including at least one switch and a plurality of links, each switch having a plurality of ports, the end node comprising;

one or more processing units, each unit having a processor and a memory; and
at least one port for connecting one or more processing units to the switching fabric;

wherein the memory of at least one of the processing units of each end node contains a fabric management process that is configured to:

obtain, from the switch, stored information and save the stored information so as to be accessible to the fabric management process, the stored information being gathered by said switch;

determine, from the stored information, whether or not a link is connected to any one of said switch ports;

determine, for each switch port having a link connected thereto, whether the stored information gathered by said switch and pertaining to each said switch port is valid;

perform, for each switch port for which the gathered information is determined to be valid, a plurality of tests on the gathered information pertaining to each said switch port;

enable, if the gathered information pertaining to each said switch port passes each test in at least a subset of the plurality of tests, each said switch port for data traffic; and

disable each said switch port for data traffic, otherwise.

14. An end node as recited in claim 13, wherein the stored information is gathered periodically by the switch.

15. An end node as recited in claim 13, wherein the fabric management process is configured to obtain, from the switch, stored information each time a recurring, prescribed interval has lapsed,

16. An end node as recited in claim 15, wherein the prescribed interval is about 60 to 180 seconds.

17. An end node as recited in claim 13, wherein the fabric management process is configured to obtain, from the switch, stored information each time the fabric management process detects that a link alive status has returned to a link that connects the end node directly to a switch.

18. A switch comprising

a plurality of ports each configured to be connected to a port of another switch or a port of an end node in a cluster;

routing hardware for routing packets from any of said plurality of switch ports to any other of said plurality of switch ports, said routing hardware including selective routing hardware control logic for enabling or disabling the transfer of data packets on each of said plurality of ports;

5 link alive hardware logic configured to allow the end nodes and switches to determine whether or not a port is connected to a live link;

an interval timer for repeatedly timing a scan interval and indicating the expiration thereof;

10 a first memory having a program resident therein that includes a routine that is operative, upon the expiration of the scan interval, to:

select, in turn, each one of said plurality of ports;

determine, for each selected port, whether or not a gather info flag is set; and

15 for each selected port connected to a live link and having the gather info flag set, construct a 'gather neighbor info request', send the constructed request over each said selected port, and receive and store any response from any port connected to each said selected port;

said program further including a routine that is operative to return, upon request, via one of said plurality of ports, all stored responses;

a processor connected to the first memory, for executing programs resident in the first memory;

20 a second memory having a configuration file resident therein, said configuration file including a routing table that specifies how packets are to be routed between said plurality of ports.

19. A switch as recited in claim 18, wherein the program resident in the first memory is further
25 operative, upon the switch being powered on, or upon the switch receiving a hard reset command from an operator, to disable all switch ports for data traffic.

20. A switch as recited in claim 18, wherein the program resident in the first memory is further
30 operative, upon the switch detecting a loss of link alive on a port, to disable said port for data traffic.

21. A switch as recited in claim 18, wherein the program resident in the first memory is further operative, immediately after the switch is powered on or when the switch receives a hard reset command from an operator, to:

select, in turn, each one of said plurality of ports;

determine, for each selected port, whether or not a gather info flag is set; and

for each selected port connected to a live link and for which the gather info flag is set, construct a 'gather neighbor info request', send the constructed request over each said selected port, and receive and store any response from any port connected to each said selected port.

22. A switch as recited in claim 18, wherein the program resident in the first memory is further operative, upon the switch detecting return of link alive on a port, to:

determine, for said port, whether or not a gather info flag is set;

if the gather info flag is set for said port, construct a 'gather neighbor info request', send the constructed request over said port, and receive and store any response from any port connected to said port.

23. A switch as recited in claim 18, wherein the configuration file in the second memory includes data parameters for each port that specify expected neighbor data values to be returned by a device connected to each particular port.

24. A switch as recited in claim 18,

further comprising an internal port configured to transfer fabric management packets; and

wherein selective routing hardware control logic is further configured to keep the internal port enabled for transferring of fabric management packets to and from any of the other switch ports, including ports that are disabled for data traffic.

25. A switch as recited in claim 18,

wherein the link alive hardware logic resides in each of said plurality of ports; and

wherein the link alive hardware logic is configured to allow the end nodes and switches to determine whether or not a port is connected to a live link by periodically sending and detecting the presence of a "keep-alive" symbol.

10055501-1224030
20200612

- 26. A switch as recited in claim 18, wherein the scan interval has a range of about 30 to 60 seconds.
- 5 27. A switch as recited in claim 18, wherein the response from said 'gather neighbor info' request includes the port number of any neighboring switch having a port connected to one of said plurality of ports.
- 28. A switch as recited in claim 18, wherein the response from said 'gather neighbor info' request includes a fabric id of any neighboring switch having a port connected to one of said plurality of ports.
- 10 29. A switch as recited in claim 18, wherein the response from said 'gather neighbor info' request includes a global unique identification number of any neighboring switch having a port connected to one of said plurality of ports.
- 30. A switch as recited in claim 18, wherein the response from said 'gather neighbor info' request includes a manufacturing part number of any neighboring switch having a port connected to one of said plurality of ports.
- 20 31. A switch as recited in claim 18, wherein the response from said 'gather neighbor info' request includes a version ID of a configuration file resident in any neighboring switch having a port connected to one of said plurality of ports.
- 25 32. A switch as recited in claim 18, wherein the response from said 'gather neighbor info' request includes a configuration tag of a configuration file resident in any neighboring switch having a port connected to one of said plurality of ports.
- 30 33. A switch as recited in claim 32, wherein the configuration tag encodes a cluster topology ID and a position ID indicating the position the switch occupies in the cluster topology.

34. A switch as recited in claim 18, wherein the response from said 'gather neighbor info' request includes a release version or version ID of a firmware program resident in any neighboring switch having a port connected to one of said plurality of ports.

35. A switch as recited in claim 18, wherein the response from said 'gather neighbor info' request includes major and minor revision numbers of a firmware program resident in any neighboring switch having a port connected to one of said plurality of ports.

36. A switch as recited in claim 18, wherein the response from said 'gather neighbor info' request includes major and minor revision numbers of a configuration file in any neighboring switch having a port connected to one of said plurality of ports.

37. A method of gathering port neighbor information in a switch having a plurality of ports, each configured to be connected to a port of another switch or port of an end node in a cluster, routing hardware for routing packets from any of said plurality of switch ports to any other of said plurality of switch ports, an interval timer for repeatedly timing a scan interval and indicating the expiration thereof, a memory having a configuration file resident therein, said configuration file including a routing table that specifies how packets are to be routed between said plurality of ports, the method comprising the steps of:

upon the expiration of the scan interval,

selecting, in turn, each one of said plurality of ports;

determining, for each selected port, whether or not the port has live link and whether or not a gather info flag is set for the port; and

for each selected port connected to a live link and having the gather info flag set,

constructing a 'gather neighbor info request', sending the constructed request over each said selected port, and receiving and storing any response from any port connected to each said selected port; and

upon receiving a request for the stored responses, returning the stored responses via one of said plurality of ports.

38. A computer readable medium having computer-executable instructions for performing a method of verifying the configuration of a switching fabric that interconnects a plurality of end nodes into a cluster, the switching fabric including at least one switch and a plurality of links, each interconnected end node having a fabric management process, each switch having a plurality of ports, the method comprising:

obtaining, from the switch, stored information and saving the stored information so as to be accessible to the fabric management process, the stored information being gathered by said switch;

determining from the stored information whether or not a link is connected to any one of said switch ports;

for each switch port having a link connected thereto, determining whether the stored information gathered by said switch and pertaining to each said switch port is valid;

for each switch port for which the gathered information is determined to be valid, performing a plurality of tests on the gathered information pertaining to each said switch port;

if the gathered information pertaining to each said switch port passes each test in at least a subset of the plurality of tests, enabling each said switch port for data traffic; and

otherwise, disabling each said switch port for data traffic.